Anabela Reigoto, *Estudante, FEUP*, André Oliveira, *Estudante, FEUP* e Baltasar Aroso, *Estudante, FEUP*

[[1]](#footnote-1)

Projeto de Processamento de Imagem e Classificação de Veículos - Carvana

*Sumário* - Este projeto consistiu no desenvolvimento e teste da manipulação de imagens fornecidas por um stand de automóveis, Carvana, através de métodos de pré-processamento de imagem, segmentação, análise e reconhecimento e classificação para o projecto final da unidade curricular Sistemas Baseados em Visão.

*Palavras-Chave* – Segmentação, Classificação, Reconhecimento.

# Introdução

E

ste relatório tem como objetivo complementar o código efetuado para o projeto final da Unidade Curricular Sistemas Baseados em Visão, elucidando a sua natureza mais teórica, visto que esta poderá não ter sido bem detalhada na apresentação.

Este trabalho tinha como primeiro objetivo a implementação de um algoritmo em MATLAB capaz de segmentar as imagens apresentadas para que se assemelhassem às mascaras *ground-truth* fornecidas, sendo o seu termo de comparação o cálculo da eficiência através do coeficiente de *Sørensen–*Dice.

Numa segunda fase, era pedida a classificação dos veículos nas suas possíveis classes: Compacto, Pick-up, Sedan ou SUV; sendo possível recorrer, para este efeito, às suas máscaras *ground-truth*. Para a realização desta tarefa havia livre-arbítrio para escolher qual a melhor disposição do carro para que o algoritmo fosse mais eficiente. Posto isto, após uma análise das imagens fornecidas escolhemos a vista lateral.

A última tarefa tinha como objetivo a classificação dos veículos na sua marca correspondente, dentro das 7 marcas possíveis: Mini, BMW, Chevrolet, Ford, Honda, Jaguar e Audi. Para a realização deste algoritmo escolhemos a disposição traseira dos automóveis, com a exceção de um Mini *Clubman* cujo símbolo da marca se encontrava no canto inferior esquerdo da traseira, o que impossibilitava a sistematização do código que procurava o símbolo na parte central da imagem, tendo sido por isso usada a disposição frontal neste caso.

# Segmentação

Numa primeira abordagem, tentou-se remover o letreiro CARVANA recorrendo a uma operação de **close** com um elemento estruturante “linha”, horizontal, grande o suficiente para que estas desaparecessem completamente (150), tentando-se de seguida proceder à *edge detection* (deteção de orla) com a função **edge**, usando o algoritmo de *Canny* e um *threshold* adequado. Os resultados ficaram aquém dos esperados, visto que, após a remoção das letras, o fundo apresentava bastantes mudanças súbitas na intensidade da cor, semelhante à imagem ter sido esboçada perfeitamente na horizontal, devido ao elemento estruturante escolhido.

Numa segunda abordagem, e percebendo desta vez que a imagem requeria algum pré-processamento mais elaborado antes de se tentar a deteção de orla, decidiu-se aplicar a operação de **close**, desta vez com uma linha ligeiramente mais curta (120), de modo a remover em grande parte as letras, mas não danificando em demasia as orlas do carro. De seguida, após aplicar um filtro gaussiano para remoção de algum ruído, efetuou-se a soma de todas as intensidades, em cada coluna da matriz obtida da leitura da imagem.[[2]](#footnote-2) Isto teve como objetivo a análise de máximos e mínimos para que se pudesse tentar encontrar as coordenadas limite do veículo, seja esse qual for. Este procedimento procura “encontrar” um retângulo que limite o veículo, tornando tudo à volta deste completamente uniforme (escolheu-se a cor cinza).

Após uma primeira observação de máximos e mínimos, encontrou-se um problema: mesmo que se recorresse a uma remoção quase completa das letras (elemento estruturante ‘linha’ de comprimento maior), as transições abruptas do fundo branco para as letras escuras permaneciam. A solução que se provou bastante eficaz consistiu numa medição das larguras de letras relevantes (‘C’, ‘R’ e ‘N’: letras que possuem fortes traços verticais, ou quase verticais), ainda na análise de máximos e mínimos. Verificou-se que, mesmo para fotografias com diferente distância à câmara fotográfica, essas larguras se encontravam em 2 intervalos restritos[[3]](#footnote-3) e facilmente detetáveis, uma vez que correspondiam sempre a um máximo precedido de um mínimo (claro; escuro; claro). A sua remoção torna-se, portanto, trivial: procuram-se os máximos e mínimos (guardados separadamente) e, sempre que houver um máximo com um mínimo precedido a uma distância aproximadamente igual à largura de uma letra, ambos são removidos. São ainda removidos todos os máximos e mínimos abaixo de um certo nível, relativo ao maior máximo. Neste momento, o algoritmo encontra-se pronto para determinar o limite esquerdo e o limite direito do veículo através do respetivo pico (máximo ou mínimo) extremo. Para encontrar os limites superior e inferior usou-se um procedimento semelhante, sendo que para estes foi possível desprezar o impacto das letras fazendo a operação de **close** com o elemento estruturante ‘linha’, comprimento 150, e aplicando um filtro gaussiano mais potente.

Tendo agora as dimensões da caixa que delimita o carro, aproximadamente, e aplicando-lhe uma margem de segurança obtida através de experimentação, colocou-se o fundo exterior a este retângulo todo a cinza. Aplica-se um algoritmo de deteção de orla de *Canny* e remove-se a orla detetada em volta do retângulo (visto que é indesejada e facilmente removível). Através da transformada de *Hough*, podemos detetar segmentos de reta na imagem correspondentes à orla das letras e a traços verticais irrelevantes, tendo em conta a sua inclinação e definindo um tamanho bom o suficiente para não apanhar traços importantes do carro, mas que apanhe orlas de letra, dentro de um limite inferior razoável. Utilizou-se o tamanho 50 e os ângulos descritíveis nos vetores [-40:-34 34:40 -26:-18 18:26] (diagonais) e [-3:3] (verticais).[[4]](#footnote-4)

No fim, com vista a “limpar” e preencher o segmento do veículo obtido, aplicou-se uma operação binária de **thin**, seguida de um **close**, depois **fill**, e por fim um **open**.

O procedimento acima descrito encontra-se dentro do script **task1.m**, na secção *Segmentation*. Esta secção efetua a segmentação de todos os veículos em modo automático, sendo que no final nos indica a eficiência média obtida através da média dos coeficientes de *Sørensen–Dice* individuais. O script faz uso de 3 funções: **startup.m** (recebe as imagens de todos os carros em escala de cinza), **getGroundTruths.m** (recebe todos os ground-truths) e **bresenham.m**. Esta última foi obtida através da biblioteca pública de funções MATLAB e retorna todos os pontos de um segmento de reta limitado pelos pontos passados como argumentos, usando uma versão do algoritmo de *Bresenham* otimizada para MATLAB (sem *loops*).

Para o cálculo dos coeficientes de *Sørensen–Dice* individuais usou-se:

(1)

Em (1), *TP* são os verdadeiros positivos, *FP* os falsos positivos e *FN* os falsos negativos.

Em anexo encontram-se os valores obtidos para a eficiência e gráficos que permitem uma boa análise quantitativa detalhada da mesma. A eficiência média obtida foi de 94.55%.

# Classificação dos veículos nas suas Classes

Os veículos da base de dados apresentada encontram-se separados em quatro classes: compactos, pick-ups, sedans e SUVs. A classificação dos veículos nas suas classes rege-se essencialmente pelas suas características dimensionais, sendo do senso comum normal de analisar que um veículo das classes pick-up e SUV é maior do que um veículo das outras duas classes. Porém, essas classificações são inconsistentes e ambíguas. Pela análise das imagens da base de dados podemos verificar que a distância dos veículos à câmara de onde foi tirada a fotografia é variável. Posto isto e para garantir uma correta classificação dos dados, fez-se uma análise pormenorizada de todas as características que, na nossa opinião, seriam preponderantes para a completa caracterização dos veículos.

Numa primeira fase, os parâmetros analisados foram: a altura, a largura, o perímetro dado pelo retângulo que delimitava a altura e a largura do veículo e a área desse mesmo retângulo (box[[5]](#footnote-5)). Foi calculado o valor máximo, mínimo e médio de todos estes parâmetros, como é apresentado na Tabela 2.

Perante a análise visual das imagens e dos resultados tabelas foi escolhida apenas a vista lateral para a classificação dos veículos.

Primeiramente procurou-se distinguir os veículos com base nas características mais intrínsecas à sua classe, como por exemplo a distinção das Pick-ups pela altura da sua suspensão ou a separação dos SUVs pela extensão da sua “cúpula” face à sua largura. Para isto foram usadas as funções **suspension.m** e **decision\_suspension.m** para a distinção das pick-ups e a função **decision\_dome.m** para a distinção dos SUVs.

A função **suspension.m** calcula o tamanho da suspensão mínima e máxima de cada classe considerando a distância entre o último pixel branco no eixo central da imagem e a altura delimitada pelos pneus do automóvel. Após a análise dos resultados apresentados na Tabela 3 verificou-se que o tamanho mínimo da suspensão dos Pick-ups era superior a todos os outros e que havia uma grande discrepância entre o tamanho da suspensão dos Pick-ups e SUVs face aos Compactos e Sedans. Posto isto, o algoritmo já conseguia separar primeiro as Pick-ups e depois as SUVs. Sendo o último parâmetro de desempate entre os Compacto se os Sedans determinado pela relação dos seus eixos vertical e horizontal, altura e largura, respetivamente. Portanto, pela divisão da altura pela sua largura observou-se, como seria de esperar dado o extenso comprimento dos Sedans, que o valor mínimo desta divisão na classe Compactos era superior ao valor máximo na classe Sedans, concluindo desta maneira a classificação de todos os veículos nas quatro classes.

Porém e devido ao facto dos valores utilizados para a separação do SUV usando o **decision\_dome.m** terem sido inseridos com base em tentativas e a margem de distinção ser muito pequena, uma outra opção para separar as SUVs seria o cálculo da sua área, usando a função do MATLAB **regionprops**, e inserindo como argumento ‘**Area’**.

Após a utilização e exploração da função **regionprops** e tendo em conta que os métodos que estavam a ser usados eram muito restritos à precisão da segmentação, procurou-se alternativas. Para isso aumentou-se a análise dos dados como é apresentado na Tabela 4. Nesta tabela são apresentados 4 parâmetros: área, formula, axis e box.

A formula corresponde ao quociente do quadrado do perímetro com a área. Este parâmetro é usado logo numa fase inicial da árvore de decisão em três dos quatro algoritmos, uma vez que possui uma grande margem de decisão, cerca de 25.80%, que distância as Pick-Ups dos restantes veículos. Este resultado é facilmente explicado tendo em conta que as Pick-Ups apresentam um tamanho elevado e uma “caixa aberta” que lhe aumenta o perímetro comparativamente aos veículos mais arredondados. A simples relação entre a soma dos catetos de um triângulo retângulo e a sua hipotenusa demonstra bem essa situação.

(2)

O axis é calculado através do quociente entre a altura e a largura e permite identificar os carros mais baixos e compridos, como os Sedans, dos carros mais altos e pequenos, como os Compactos.

(3)

A box é calculada pelo produto da altura com a largura, correspondendo à área do retângulo definido pelo *BoundingBox*. Este parâmetro é usado apenas no primeiro algoritmo para a separação entre os Sedans e os SUVs, dado o seu porte face à pequena altura dos Sedans e uma largura semelhante.

(4)

Depois de uma análise detalhada dos dados obtidos e procurando criar uma árvore de decisão robusta aos erros da segmentação foram calculadas as margens de decisão tendo em conta os seis parâmetros apresentados na Tabela 5.

Perante as combinações nessa tabela assinaladas criou-se quatro algoritmos diferentes para a determinação da classe dos veículos. Todos os algoritmos apresentaram uma eficiência de 100% usando as imagens *ground-truth* na vista lateral.

Na utilização do algoritmo para a classificação das imagens obtidas com a segmentação da tarefa 1 obtivemos as eficiências apresentadas na Tabela 6. A eficiência mais elevada foi de apenas 88% no Algoritmo 1, sendo que todos os algoritmos demostraram uma má classificação na classe das Pick-Ups. Isto deve-se ao facto do critério de decisão entre a separação dos Sedan com os Compactos ser a relação da altura com a largura e a sua margem de decisão apresentada na Tabela ser baixa, cerca de 5%. Para além disso, como se pode verificar na análise dos resultados das eficiências pelas classes da Tabela 1, os Sedan apresentam a eficiência mais baixa o que afeta mais a classificação, dado que em todos os algoritmos estes são separados por margem de decisão baixas.

Primeiramente a intenção da criação de quatro algoritmos seria aumentar a robustez da classificação fase às imperfeições da segmentação, porém, após uma análise dos resultados obtidos verificou-se que nenhuma combinação seria melhor do que o Algoritmo 1, uma vez que todas as combinações erravam nos mesmos aspetos, a separação dos Compactos com os Sedans, e a falha na classificação das Pick-Ups.

# Classificação dos veículos nas suas Marcas

Os veículos da base de dados fornecida apresentam sete marcas diferentes: Mini, BMW, Chevrolet, Ford, Honda, Jaguar e Audi. A classificação dos veículos nas suas marcas baseia-se na comparação do símbolo presente na traseira do carro com os símbolos das diferentes marcas. Para tal foi criada uma base de dados com os símbolos das sete marcas.

Na análise das imagens, podemos verificar que na traseira do carro, o símbolo da marca encontra-se maioritariamente no centro, à exceção do Mini *Clubman,*tendo esta localização pequenas variações entre diferentes classes de veículos.

Primeiramente, utilizou-se o algoritmo desenvolvido na tarefa 2 para distinguir os veículos em classes, utilizando as imagens *ground-truth*. Posteriormente selecionou-se a janela de corte consoante a classe do veículo em causa. Para o pré-processamento da imagem foi usado o filtro Laplaciano de Gauss com um sigma de 0.4 para as imagens a classificar e de 0.5 para as imagens da base de dados criada para comparação. Com isto procurou-se delimitar as linhas que definem o logótipo das marcas para que a correlação fosse mais precisa.

Na análise da correlação observou-se que os valores oscilam muito de marca para marca, sendo esta a razão de uma eficiência de 62.5%. O facto de alguns logotipos, nomeadamente o do Chevrolet, apresentarem uma correlação média superior na maioria dos carros, influencia a decisão do máximo valor da correlação.

Para uma maior uniformização dos resultados da convolução tendo em conta o mesmo símbolo para diferentes carros, tentou-se melhorar a classificação ao calcular o máximo valor da correlação de cada símbolo para os 16 carros e dividir esse vetor por esse máximo, procurando normalizar os valores. Porém o resultado obtido não foi o pretendido e o valor da eficiência diminuiu para 43.75%. Depois da análise dos resultados percebeu-se que vetores de correlações com valores menos distantes pioram a classificação. Por exemplo, considerando dois vetores, um com um primeiro máximo muito maior que o segundo, que resulta numa normalização em que o segundo máximo é por exemplo 0.5; e outro vetor onde há um conjunto de máximos relativos próximos do máximo absoluto, cuja normalização se sobrepõe aos 0.5 apesar desse não ser a escolha acertada.

# Conclusion

## Em suma, o processo mais complexo do projeto baseou-se na segmentação das imagens, que exigiu um maior teste e pré-processamento dos veículos. A classificação é, neste caso, muito influenciada pela segmentação por ela antecedida, sendo que deve ter por base o seu estudo na criação da árvore de decisão. Sendo certo que a eficiência dos algoritmos apresentados seria muito maior caso estes mesmos fossem feitos tendo em conta a análise das imagens obtidas pela segmentação, visto que esta pode cortar características determinantes do veículo. A última tarefa remete para uma classificação de outro tipo, mas que é fundamentada nas duas tarefas anteriores para que haja um resultado mais eficaz. Esta tem por base os resultados de uma correlação que falhou pela falta de pré-processamento e uma pouca consistência na disposição dos logotipos no centro das imagens.

Appendix







# 

# 

# 





References

Rafael C.Gonzalez, Richard E.Woods, Steven L. Eddins, *Digital Image Processing Using Matlab*. New Jersey, United States Of America, 2004

1. [↑](#footnote-ref-1)
2. Todas as imagens lidas pelo algoritmo em MATLAB foram, aquando da sua leitura, convertidas de RGB para escala de cinza (rgb2gray) e para dupla precisão (im2double). [↑](#footnote-ref-2)
3. Presentes no código MATLAB nas variáveis *lw\_sup* e *lw\_inf*, que contêm, respetivamente, os limites superiores e inferiores desses intervalos. Os valores que serviram de guia para esses intervalos encontram-se acima dessas variáveis, comentados. [↑](#footnote-ref-3)
4. Os ângulos das letras foram medidos com recurso à ferramenta de visualização de imagem do MATLAB e através de conhecimento básico matemático, analisando pontos dessas mesmas letras. [↑](#footnote-ref-4)
5. Este será o nome utilizado para a área do retângulo calculada pelo produto da altura com a largura em contraste com a área posteriormente mencionada como a soma de todos os pixéis brancos na imagem do ground-truth. [↑](#footnote-ref-5)